Simulation of Quantum Algorithms for Solving Machine Learning and Chemistry Problems

Steven Oud
SURFsara, Amsterdam

October 2, 2019

Summary

Quantum computing is a promising technology of the near future. Since its first proposal researchers have been looking to use it for solving difficult chemistry and physics problems. More recently, research towards its application in the field of machine learning has shown promise. We take a look at how to apply current state-of-the-art quantum algorithms to solve machine learning and chemistry problems.

Contents

1 Introduction

Quantum computing is one of the most promising technology developments of the coming years. Big companies like IBM (IBM, 2019), Google (Google AI, 2019), Intel (Intel, 2019), Microsoft (Microsoft, 2019), and countries like the USA and China are investing huge amounts of money into the development of quantum computers (Raymer and Monroes, 2019; Kania and Costello, 2018). The development of practical quantum computers that can be used to solve real-world problems is driven by the expectation that for certain tasks, quantum computers can dramatically outperform classical computers (Preskill, 2012).

Already last year, SURFsara started a number of activities and collaborations in the field of quantum computing. The overall objective of SURFsara is to support Dutch researchers in taking an early and competitive advantage in quantum computing technologies and facilities while these become available. Like with any emerging compute technology, it needs early adopters in the scientific computing community to identify problems of practical interest that are suitable as proof-of-concept applications. In the context of the SURF Open Innovation Lab, SURFsara seeks to stimulate and support these advances in scientific research in close collaboration with research groups. Within this context SURFsara is interested in testing, benchmarking and creating good examples of quantum computing applications that can be used by the scientific community.

SURFsara HPC center supports various research institutes and universities of the Netherlands. The chemistry community is currently one of the largest; the machine learning community probably the fastest growing one. Both of these fields are large areas of research within the quantum computing field, expecting improvements to be gained from them. As the main interest of SURFsara is to support the potential main use cases of quantum computers in the future, the tasks of this internship will be focused on quantum machine learning (QML) and quantum chemistry (QC).

This report will try to give an answer to the main question "*How can quantum algorithms be implemented using simulated quantum circuits to solve machine learning and chemistry problems?*" This is further expanded into the following sub questions:

1. What are current promising QML and QC algorithms?
2. What simulators are best suited for simulating QML and QC quantum circuits?
3. How can these quantum algorithms be integrated in current classical workflow of chemistry and machine learning applications?
4. How do these quantum algorithms perform (in regard to classical algorithms)?

This report is structured as follows. First, in Section 2, we describe how the research was conducted. Second, in Section 3, a short introduction to quantum computation and information is given. Third, in Sections 4, 5, and 6, an overview of the (quantum) machine learning and quantum chemistry fields is given. Finally, in Section 7, quantum algorithms are implemented using simulation and the results are discussed. The conclusion of the research is described in Section 8.

2 Methods

The research in this report was done mostly through desk research, with the help and advice of my colleagues. The first step was to get familiar with the QML and QC fields; what background knowledge is required, what is the current state of research and what are the next steps? To get started, several papers in promising areas of research were provided by colleagues. From there, databases such as Google Scholar and arXiv were used to find further information about these areas. Search terms used include *quantum machine learning*, *quantum neural network*, *quantum support vector machine*, *variational quantum eigensolver*, *quantum phase estimation*, *quantum perceptron*, *quantum classifier*, *quantum machine learning library*, *quantum chemistry simulation* and *hybrid quantum optimization*.

3 Quantum Computation and Information

Quantum computers take advantage of quantum mechanical effects such as superposition and entanglement to solve certain problems faster than classical computers. The idea of a quantum computer was first proposed by Richard Feynman for solving problems in physics and chemistry (Feynman, 1982). Since then the field has advanced a lot with algorithms such as Shor's algorithm for factoring integers (Shor, 1999) and Grover's search algorithm (Grover, 1996). These quantum algorithms show efficient solutions for problems that are considered hard for classical computers.

This section summarizes basic concepts of quantum computation and information theory required to understand how quantum computers can be used to speedup certain computational problems. This is a very high level overview of quantum computation and is far from a complete introduction to the field. For a more complete introduction to quantum computation and information, we refer to the book by Nielsen and Chuang, 2011.

3.1 Quantum States

The basic and smallest unit of quantum information is the quantum bit, or *qubit*. A qubit is a two-state quantum-mechanical system, which can be represented for example in the spin of an electron (but any quantized physical property can be used). A qubit can be in a state of an orthonormal basis $\{|0\rangle, |1\rangle\}$ (corresponding to spin up and spin down for the electron example), but it can also be in a linear combination, or *superposition*, of states. The state of a qubit (the so-called *wave function*) can be described as following:

$$|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle . \tag{1}$$

Quantum states are denoted using Dirac notation $|\cdot\rangle$, which is a column vector in $\mathbb{C}^n$. Here the coefficients $\{c_0, c_1\}$ are complex valued numbers called *probability amplitudes*. When we measure a quantum state, it collapses probabilistically to one of the basis states. The norm square $|c_0|^2$ gives us the probability of finding the particle in state $|0\rangle$, and $|c_1|^2$ gives us the probability of finding the particle in state $|1\rangle$. As we are dealing with probabilities, the quantum state should be normalized: $\sum_{i=0} |c_i|^2 = 1$. As an example we have an arbitrary state:

$$|\psi\rangle = \frac{i}{2} |0\rangle - \frac{\sqrt{3}}{2} |1\rangle . \tag{2}$$

After measuring this state, we have a $|i/2|^2 = 1/4$ chance of the system being in the state $|0\rangle$, and a $|-\sqrt{3}/2|^2 = 3/4$ chance of being in the state $|1\rangle$. The laws of quantum mechanics greatly restrict our access to information stored in quantum states. We cannot access the amplitudes of a state, other than by preparing and sampling a state many times to get an approximation. When we measure a state, it collapses to a basis state $|j\rangle$ with probability $|c_j|^2$. If we measure again immediately after the first measure, we get the same result 100% of the time. So if we measure $|1\rangle$ and measure again immediately after, we will see $|1\rangle$ again. Measuring a quantum state collapses the wave function and destroys state information.

We define the computational basis states $\{|0\rangle, |1\rangle\}$ as following:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{3}$$

This idea generalizes to multi-qubit systems. A multi-qubit system with n states $\{|\psi_1\rangle, |\psi_2\rangle, \ldots, |\psi_n\rangle\}$ can be represented using the *Kronecker product*[1]:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle , \tag{4}$$

[1] Entangled states are an exception to this, which will be further discussed in Section 3.3

resulting in a $N = 2^n$ dimensional state $|\psi\rangle$. For example, a three qubit state lives in a 2^3-dimensional Hilbert space spanned by computational basis states $\{|000\rangle, |001\rangle, |010\rangle, \ldots, |111\rangle\}$:

$$\begin{aligned} |\psi\rangle &= c_0 |000\rangle + c_1 |001\rangle + c_2 |010\rangle + \cdots + c_{N-1} |111\rangle \\ &= c_0 \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + c_1 \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + c_2 \begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} + \cdots + c_{N-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{N-1} \end{pmatrix}. \end{aligned} \tag{5}$$

3.2 Quantum State Evolution

In classical computers, we manipulate bits through logical gates. Equivalently, quantum computers use quantum gates, which transforms one quantum state to another through an operator U. These operators must be unitary. That is, they must preserve the norm of the vector and be reversible: $U^\dagger U = UU^\dagger = I$ (where † is the complex conjugate and I the identity matrix). Single qubit states can be thought of as a vector on the surface of a sphere (the Bloch sphere). A unitary operation can then be thought of as rotations around the x, y and z axes of the sphere (Figure 1).

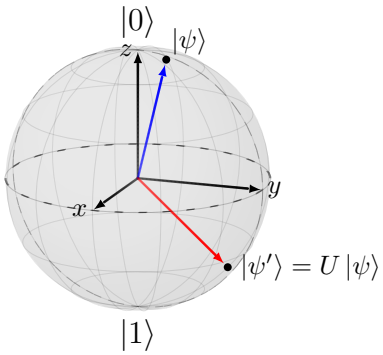


Figure 1: *Arbitrary transformation of state $|\psi\rangle$ by operator U visualized on the Bloch sphere.*

As an example, the X gate is equivalent to the classical NOT: $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$. It is known as one of the three Pauli matrices used in quantum mechanics:

$$\sigma_1 = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{6}$$

Another frequently used gate is the Hadamard gate, which produces an equal superposition:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{7}$$

It acts on the computational basis states as following:

$$\begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \\ H|1\rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{aligned} \tag{8}$$

Applying the Hadamard gate again on the equal superposition created above is an example of destructive interference:

$$\begin{aligned} H\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) &= |0\rangle\,, \\ H\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) &= |1\rangle\,. \end{aligned} \tag{9}$$

We find that applying the Hadamard twice is the same as doing nothing, or more formally, we say the Hadamard gate is *Hermitian*: $H = H^\dagger$.

3.3 Entanglement

There also exist multi-qubit gates, which are required for universal quantum computing. These gates introduce the phenomena entanglement. Consider the following state:

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{10}$$

Measuring the first qubit gives state $|0\rangle$ with probability $1/2$, and state $|1\rangle$ with probability $1/2$. However, the measurement immediately collapsed the whole state to either $|00\rangle$ or $|11\rangle$. So by measuring the first qubit, we also know the state of the second qubit. The individual states are related to each other, and this relation is called entanglement. More formally, two qubits are entangled if and only if the state of those two qubits cannot be expressed as two individual states. For the entangled state $|\Phi^+\rangle$, let's assume there exist amplitudes $\{a, b, c, d\}$ such that:

$$\left|\Phi^+\right\rangle = (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{11}$$

However, this would imply that $ac = bd = 1/\sqrt{2}$ and $ad = bc = 0$, which is a contradiction.

The remarkable thing about entanglement is that it works over any distance. That is, given the state $|\Phi^+\rangle$, one qubit could be located in another galaxy while the other qubit is located on earth. When we measure the qubit on earth, we immediately know that the qubit in the other galaxy is in the same state as we measured. This does not allow for faster than light communication, as a classical channel is still required to communicate results between the observers.

3.4 Quantum Speedup

Quantum algorithms promise to provide a speedup over classical algorithms by "abusing" quantum phenomena such as superposition, interference and entanglement. They usually offer a quadratic or exponential speedup over their classical counterpart. Aaronson, 2011 probably described it best: "The goal in quantum computing is to choreograph a computation so that the amplitudes leading to wrong answers cancel each other out, while the amplitudes leading to right answers reinforce."

The most famous example of a quantum speedup is probably Shor's algorithm for factoring integers and computing discrete logarithms, which was introduced in 1994. Its implications are huge for modern cryptography algorithms such as RSA, which depend on the fact that factoring integers is a hard problem on classical computers. Shor showed that this can be done in polynomial time on a quantum computer (Shor, 1999), providing an exponential speedup over classical algorithms. It has been estimated that a 2048-bit RSA integer could be factored in eight hours using 20 million noisy qubits (Gidney and Ekerå, 2019), which we are very far away from achieving (companies are currently struggling to have 50 coherent qubits).

There is some very convincing theoretical evidence that quantum computers can speed up certain computations by a large factor. Till we have reached quantum supremacy though, it is just that: theoretical. We should however keep working with the resources we currently have available to advance the field of quantum computing and push the limitations of computation.

3.5 Current State of Quantum Computing

It is important to note that quantum computers are still at the experimental stage, and a lot of results are theoretical. For example, the highest number factored by using Shor's algorithm on an actual quantum computer thus far is 21 (Martín-López et al., 2012). Furthermore, quantum supremacy, which is the potential ability of a quantum computer to solve problems that classical computers practically cannot (Preskill, 2012), has not been achieved yet as of writing this. Current and near future quantum computers—often referred to as noisy intermediate-scale quantum (NISQ) devices—will have about 50-100 qubits and may be able to achieve quantum supremacy (Preskill, 2018). However, they suffer from noise, which greatly limits their coherence time and thus usefulness. Quantum error correcting codes exist which can be used to protect from noise, but these require a high amount of qubits per logical qubit.

Research has adapted to the limitations of NISQ devices, and hybrid quantum/classical algorithms have become an important area of research. These hybrid algorithms are often referred to as *variational quantum algorithms*. Examples of such algorithms are the quantum approximate optimization algorithm (QAOA) for optimization problems (Farhi, Goldstone, and Gutmann, 2014), and the variational quantum eigensolver (VQE) for finding eigenvalues of a Hamiltonian (Peruzzo et al., 2014). These variational algorithms typically involve a small quantum subroutine run inside of a classical optimization loop, removing the need for a large-scale, coherent quantum computer.

So far we have been talking about so-called universal gate model quantum computers. There exist special purpose quantum computers called quantum annealers, which have achieved qubit counts up to 2000 (Gibney, 2017). These quantum annealers are restricted to solving optimization problems, of which its usefulness has been doubted (Shin, Smith, Smolin, and Vazirani, 2014; Aaronson, 2013; Rønnow et al., 2014). This research is focused on universal gate model quantum computers, as its usefulness and applications are more evident (although not certain).

4 Classical Machine Learning

Before looking at quantum machine learning, we first explore classical machine learning techniques, which quantum machine learning is largely built upon. Instead of being explicitly programmed to perform a certain task, a machine learning algorithms learns by analyzing large amounts of data. As a subfield of artificial intelligence, machine learning is used to perform complex tasks which come natural to us humans, such as image recognition, spam detection and recommendation.

Machine learning algorithms are often divided into three categories: *supervised*, *unsupervised* and *reinforcement learning*. In supervised learning, we have a predefined set of training data in which data points are correctly labeled. Its goal is then to predict the correct label for new data it has not seen. In unsupervised learning we do not have labeled data points, and instead we try to find previously unknown patterns in the data set. Finally, in reinforcement learning, an agent learns how to behave in an environment by interacting with it and receiving feedback based on these actions.

Artificial neural networks (ANN) are ubiquitous in the field of supervised machine learning. They are inspired by the biological neural networks found in our brains and have developed into a wide range of methods that have advanced the state of the art in various fields. Some of the methods include feedforward neural networks for classification and regression, convolutional neural networks for processing visual data (Cireşan, Meier, and Schmidhuber, 2012) and generative adversarial networks for generating new data based on existing data (Goodfellow et al., 2014). Another notable machine learning model for supervised learning is support vector machine (SVM), which can be used for classification or regression. They have been found to be able to solve problems such as semantic parsing (Pradhan, Ward, Hacioglu, Martin, and Jurafsky, 2004) and handwritten character recognition (Decoste and Schölkopf, 2002).

A popular method of unsupervised learning is clustering. Clustering is the task of grouping data points together with other data points which are most similar. Some well-known clustering algorithms include k-means clustering and hierarchical clustering. Generative adversarial networks

can be thought of as unsupervised learning algorithms and can even be used in reinforcement learning (Ho and Ermon, 2016).

Classical machine learning algorithms are well defined and can be used to classify images, recognize patterns and speech, play complex games and much more. However, as the amount of globally stored data is growing by about 20% every year (Hilbert and López, 2011), researchers are looking more and more for innovative approaches to machine learning. A promising idea that is currently being investigated is the use of quantum computers to optimize classical machine learning algorithms (Schuld, Sinayskiy, and Petruccione, 2015). The research field of quantum machine learning may offer new ways of intelligent data processing and even guide and advance classical machine learning.

5 Quantum Machine Learning

It has been known for a while that for some problems quantum algorithms can offer a speedup over their classical counterpart (Nielsen and Chuang, 2011). Recent advances in quantum computing suggest that machine learning algorithms may also be susceptible to a quantum speedup (Lee et al., 2019; Lloyd, Mohseni, and Rebentrost, 2013; Gao, Zhang, and Duan, 2018). Quantum machine learning is still in its very first stage of development, and lacks a widely agreed upon definition. However, there have been promising developments in the recent years. An efficient quantum algorithm named HHL (after its inventors) for solving a system of linear equations was found by Harrow, Hassidim, and Lloyd, 2009. In the years since, HHL has been proposed for multiple machine learning applications[2], such as k-means clustering (Lloyd et al., 2013) and support vector machines (Rebentrost, Mohseni, and Lloyd, 2014).

HHL-based algorithms require quantum computers capable of executing high-depth circuits, which we won't have access to any time soon. To circumvent this requirement, variational QML algorithms have become a main focus area of research. Variational QML algorithms have shown the ability to perform machine learning tasks on NISQ devices while providing a means to deal with high dimensional data, which has been unpractical on classical computers (Mitarai, Negoro, Kitagawa, and Fujii, 2018). Promising results have been shown with quantum neural networks (Farhi and Neven, 2018; Schuld, Bocharov, Svore, and Wiebe, 2018; Fanizza, Mari, and Giovannetti, 2019), quantum generative adversarial networks (Romero and Aspuru-Guzik, 2019; Benedetti, Grant, Wossnig, and Severini, 2019) and quantum support vector machines (Havlicek et al., 2019; Schuld and Killoran, 2019).

Quantum machine learning can help us process the huge amounts of data we keep creating, due to their ability to encode exponentially more data in their state compared to classical bits. Furthermore, they can potentially speed up certain computational heavy parts used in machine learning algorithms by a large factor. Finally, we can use QML algorithms to learn properties about quantum states, for which exists no classical equivalent.

To realize these applications we need to overcome some major obstacles. First, we need to build quantum computers with more qubits, better qubits and longer coherence times. Second, we need an efficient way to prepare classical data into a quantum state. A quantum random access memory (qRAM) architecture has been proposed for this (Giovannetti, Lloyd, and Maccone, 2008), but it largely depends on the data being uniform, which our data often is not. Third, similarly to the previous point, is the readout problem. Since measurement destroys a quantum mechanical system's wave function, we need to do repeated measurements to extract any meaningful information of the system. This implies we can't efficiently retrieve output data stored in an exponentially large Hilbert space.

[2]HHL and the algorithms which use it can only provide an exponential speedup by overcoming some major caveats (Aaronson, 2015). For example, the problem of loading a large amount of classical data into a quantum computer can quickly diminish or destroy the potential quantum speedup.

6 Quantum Chemistry

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce lobortis erat eget erat euismod dapibus. Interdum et malesuada fames ac ante ipsum primis in faucibus. Donec pharetra, magna ac tincidunt varius, neque odio lobortis velit, eget tincidunt lectus sapien ac velit. Donec maximus euismod arcu, at efficitur urna. Praesent viverra elementum elementum. Cras lacinia nisi eleifend sodales posuere. Fusce ut blandit purus, quis ornare est. Aenean varius purus lorem, a lobortis orci mollis pretium. Nulla facilisi. Nulla in neque orci. Integer dolor massa, ullamcorper nec sagittis id, mattis a lectus. Vivamus efficitur mi a elit feugiat facilisis. Ut lorem tortor, dignissim nec quam et, dapibus vehicula nibh. Quisque mollis enim quis odio interdum, ac blandit mauris maximus. Aenean ac dolor in augue accumsan porta.

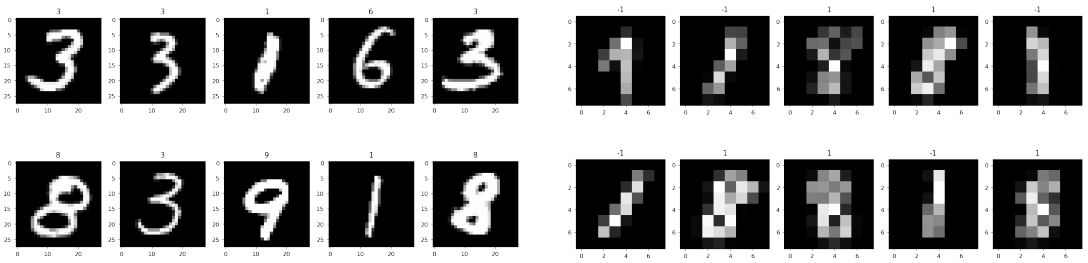
Fusce at sodales turpis. Nullam pulvinar rhoncus diam eu consequat. Sed et ante ac augue rutrum cursus id id mi. Quisque aliquet at lectus eget condimentum. Quisque pharetra nulla vel vestibulum vulputate. Donec blandit lacus est, sed tempor nisi laoreet eget. Mauris maximus ultricies varius. Nullam id hendrerit ante, quis efficitur nibh. Pellentesque convallis dignissim dapibus. Nunc vehicula scelerisque posuere.

7 Implementations and Results

This section describes how QML and QC algorithms were implemented, tested and benchmarked. The experimental results described were achieved by simulating quantum circuits using SURFsara's Lisa computing system (SURFsara, 2019). Simulations were executed using the PennyLane (Bergholm, Izaac, Schuld, Gogolin, and Killoran, 2018) framework.

7.1 Quantum Neural Network

A quantum neural network (QNN) inspired by the work of Farhi and Neven, 2018 was implemented for classifying handwritten digits. The famous MNIST database of handwritten digits (LeCun and Cortes, 2010) was used as data set (Figure 2). For initial experiment, the images of the data set were downsampled to a lower dimension and the task at hand was limited to binary classification of digits 1 and 8 to simplify the problem and reduce resource requirements. After a successful initial experiment, the experiment was extended to one-vs-rest classification of the original 28×28 data set, and finally generalized to multi-class classification of the original data set.



(a) *Original data set of* 28×28 *images.* ***(b)*** *Downsampled binary data set of* 8×8 *images.*

Figure 2: *Samples of MNIST data set used for binary classification.*

7.1.1 Binary Downsampled Classifier

The problem of supervised classification in machine learning is defined as following: given a training set of data whose category is known, train a model which is able to identify to which category new data belongs. A simple example that we will tackle in this section is recognizing if a handwritten digit is a 1 or an 8.

We implement a QNN to classify 8×8 images of handwritten digits (Figure 2b). It uses a quantum circuit together with a classical optimization algorithm to find the optimal parameters for the quantum circuit, known as a variational algorithm. An overview of the quantum circuit used in the experiment can be found in Figure 3. The variational QNN classifier works as following:

1. Prepare data sample $\vec{x}$ in the amplitudes of a quantum state $|x\rangle$.
2. Prepare a readout qubit in the state $|1\rangle$, giving $|x, 1\rangle$.
3. Apply L layers of parameterized unitaries: $U(\vec{\theta})\,|x, 1\rangle$, where $U(\vec{\theta}) = U(\vec{\theta}_L)U(\vec{\theta}_{L-1})\dots U(\vec{\theta}_1)$.
4. Measure expectation value $\langle\sigma_z\rangle = \langle x, 1|\, U(\vec{\theta})^\dagger \sigma_z U(\vec{\theta})\,|x, 1\rangle$ of readout qubit.
5. Calculate and minimize $\text{loss}(\vec{\theta}, \langle\sigma_z\rangle)$ through a classical optimization algorithm.
6. Repeat until convergence.

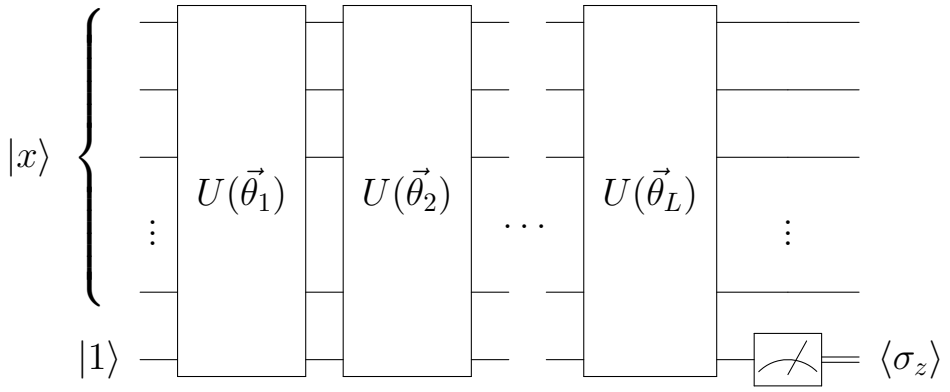


Figure 3: *General circuit of the QNN. The network consists of L layers with each layer implementing a parameterized unitary $U(\vec{\theta}_l)$ whose parameters are optimized using a classical optimization algorithm. At the end of the circuit, the expectation value $\langle\sigma_z\rangle$ of the readout qubit is measured and the predicted class is decided by $\text{sgn}(\langle\sigma_z\rangle)$. The decomposition of a layer is shown in Figure 4.*

We trained a QNN using a training set of 2014 images and a test set of 843 images. As optimizer, we used mini-batch stochastic gradient descent (SGD) with a learning rate of 0.4 and a batch size of 32. After training for 20 epochs with 2 layers, it managed to reach 91.1% accuracy without any hyperparameter tuning or circuit optimization. The goal of this experiment is not to compete with classical state-of-the-art neural networks, which are capable of achieving an error percentage of only 0.23% (Cireşan et al., 2012)[3]. Rather, it is meant to demonstrate the possibility of using quantum computers for machine learning problems and to hint that the universal approximation theorem (Csáji, 2001) applies to quantum neural networks. It shows the potential, given we solve the issues surrounding qRAM (Aaronson, 2015), to use quantum computers to process (quantum) data which classical computers cannot process efficiently.

7.1.2 One-vs-Rest Classifier

To make a step into the direction of multi-class classification using a QNN, we first consider the problem of one-vs-rest classification. We use the original 28×28 MNIST data set (Figure 2a) to

[3]Which is not to say quantum neural networks can't compete with classical neural networks. In fact, Pérez-Salinas, Cervera-Lierta, Gil-Fuster, and Latorre, 2019 show a single-qubit classifier to be at least comparable with classical methods.

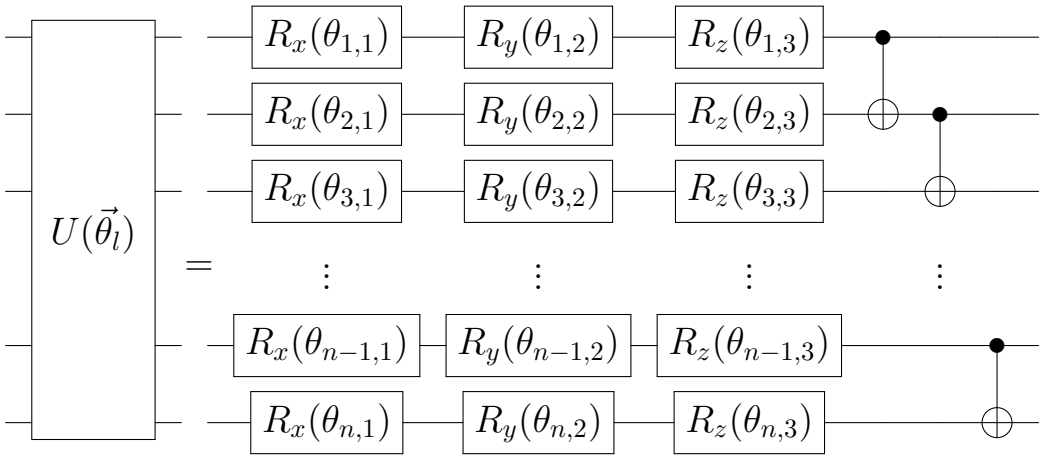


Figure 4: *Decomposition of a layer $U(\vec{\theta}_l)$. The amount of trainable parameters for a QNN with n qubits and L layers is $3n \cdot L$. The amount of qubits required is decided by the dimension of the data: $n = \lceil \log_2 N \rceil + 1$, where N is the input dimension. The downsampled MNIST classifier requires $\lceil \log_2(8 \cdot 8) \rceil + 1 = 7$ qubits.*

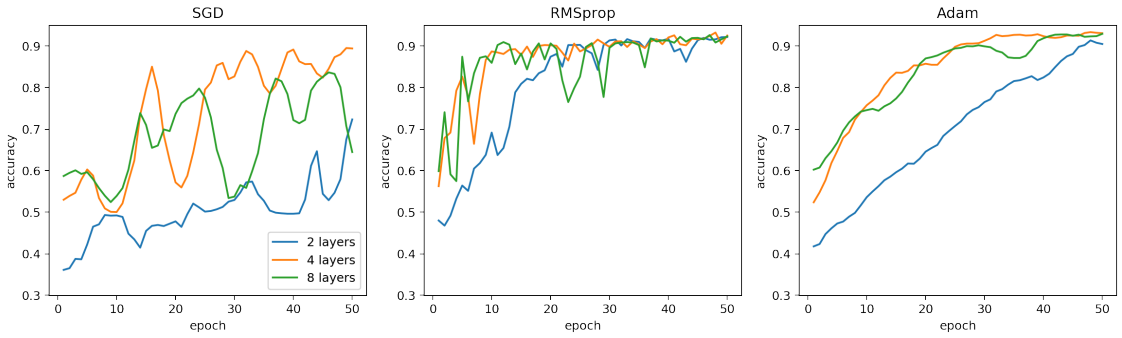
train a QNN to detect if a handwritten digit is 0 or not 0. To encode a 28×28-dimensional image we need $\lceil \log_2(28 \cdot 28) \rceil = 10$ qubits, plus a readout qubit, for 11 qubits total. As the complexity of the problem grows, we compare the performance of multiple optimizers: SGD, RMSprop and Adam. We used a set of training data with 5923 samples, and a set of test data with 1960 samples. Some hyperparameter tuning was done to find the optimal learning rate, but more tuning can be done to possibly increase performance. The results of the experiment can be found in Figure 5. Increasing the amount of layers seems to improve convergence rate, but ultimately there doesn't seem to be a big performance difference between 4 and 8 layers. As the feature space grows, SGD has trouble converging, resulting in a sweet spot of 4 layers where it performs best, but it is still easily outperformed by the other optimizers. RMSprop and Adam both seem to converge to 92.5% accuracy with 4 and 8 layers, and RMSprop even manages this with 2 layers.

7.1.3 Multiclass Classifier

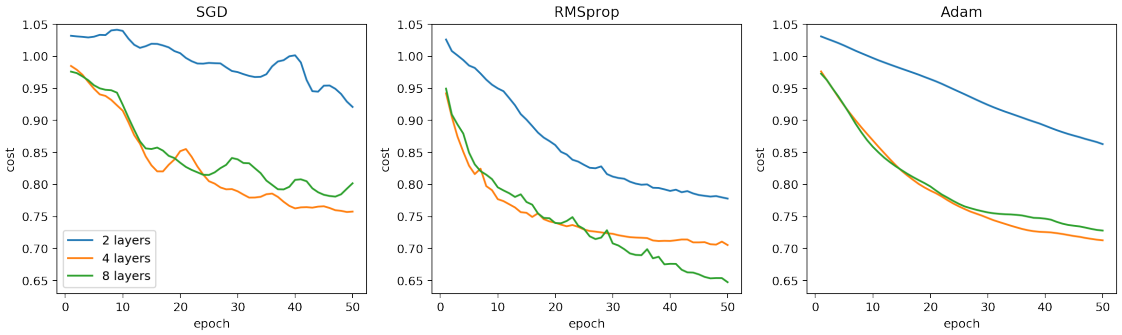
Finally we generalize the QNN to a multiclass classifier by implementing a fidelity loss function as proposed by Pérez-Salinas et al., 2019. To implement this we need to make some changes to the QNN described in Section 7.1.1. First, we introduce C maximally orthogonal label states, where C is the number of classes. Then, instead of using the square loss function on the expectation value as cost, we measure the fidelity of the final state of the readout qubit and the label state. The goal then is to maximize the fidelity between the final state of the readout qubit and the correct label state.

We conduct a similar experiment to that from the previous section. The problem at hand is to distinguish between the handwritten digits 1 through 6 including, giving six classes.

8 Conclusion



(a) *Accuracy on test data set with different optimizers.*



(b) *Cost landscape of different optimizers.*

Figure 5: *Experimental results of a one-vs-rest QNN classifier for detecting if a handwritten digit is 0 or not.*

References

Aaronson, S. (2011). Quantum computing promises new insights, not just supermachines. Retrieved 2019-09-18, from https://www.nytimes.com/2011/12/06/science/scott-aaronson-quantum-computing-promises-new-insights.html

Aaronson, S. (2013). D-Wave: Truth finally starts to emerge. Retrieved 2019-09-19, from https://www.scottaaronson.com/blog/?p=1400

Aaronson, S. (2015). Read the fine print. *Nature Physics*, *11*(4), 291.

Benedetti, M., Grant, E., Wossnig, L., & Severini, S. (2019). Adversarial quantum circuit learning for pure state approximation. *New Journal of Physics*, *21*(4), 043023.

Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., & Killoran, N. (2018). Pennylane: Automatic differentiation of hybrid quantum-classical computations. arXiv: 1811.04968 [quant-ph]

Cireşan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. arXiv: 1202.2745 [cs]

Csáji, B. C. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, *24*, 48.

Decoste, D. & Schölkopf, B. (2002). Training invariant support vector machines. *Machine learning*, *46*(1-3), 161–190.

Fanizza, M., Mari, A., & Giovannetti, V. (2019). Optimal universal learning machines for quantum state discrimination. *IEEE Transactions on Information Theory*.

Farhi, E., Goldstone, J., & Gutmann, S. (2014). A quantum approximate optimization algorithm. arXiv: 1411.4028 [quant-ph]

Farhi, E. & Neven, H. (2018). Classification with quantum neural networks on near term processors. arXiv: 1802.06002 [quant-ph]

Feynman, R. P. (1982). Simulating physics with computers. *International journal of theoretical physics*, *21*(6), 467–488.

Gao, X., Zhang, Z.-Y., & Duan, L.-M. (2018). A quantum machine learning algorithm based on generative models. *Science Advances*, *4*(12). doi:10.1126/sciadv.aat9004. eprint: https://advances.sciencemag.org/content/4/12/eaat9004.full.pdf

Gibney, E. (2017). D-Wave upgrade: How scientists are using the world's most controversial quantum computer. *Nature News*, *541*(7638), 447.

Gidney, C. & Ekerå, M. (2019). How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. arXiv: 1905.09749 [quant-ph]

Giovannetti, V., Lloyd, S., & Maccone, L. (2008). Quantum random access memory. *Physical review letters*, *100*(16), 160501.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Google AI. (2019). Quantum. Retrieved 2019-09-11, from https://ai.google/research/teams/applied-science/quantum/

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. arXiv: quant-ph/9605043

Harrow, A. W., Hassidim, A., & Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Physical review letters*, *103*(15), 150502.

Havlicek, V., Corcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature*, *567*(7747), 209.

Hilbert, M. & López, P. (2011). The world's technological capacity to store, communicate, and compute information. *science*, *332*(6025), 60–65.

Ho, J. & Ermon, S. (2016). Generative adversarial imitation learning. In *Advances in neural information processing systems* (pp. 4565–4573).

IBM. (2019). Quantum computing. Retrieved 2019-09-11, from https://www.ibm.com/quantum-computing/

Intel. (2019). Quantum computing. Retrieved 2019-09-11, from https://www.intel.com/content/www/us/en/research/quantum-computing.html

Kania, E. B. & Costello, J. (2018). Quantum hegemony? China's ambitions and the challenge to U.S. innovation leadership frontier. Retrieved 2019-09-16, from https://www.cnas.org/publications/reports/quantum-hegemony

LeCun, Y. & Cortes, C. (2010). MNIST handwritten digit database. Retrieved from http://yann.lecun.com/exdb/mnist/

Lee, J.-S., Bang, J., Hong, S., Lee, C., Seol, K. H., Lee, J., & Lee, K.-G. (2019). Experimental demonstration of quantum learning speedup with classical input data. *Physical Review A*, *99*(1), 012313.

Lloyd, S., Mohseni, M., & Rebentrost, P. (2013). Quantum algorithms for supervised and unsupervised machine learning. arXiv: 1307.0411 [quant-ph]

Martín-López, E., Laing, A., Lawson, T., Alvarez, R., Zhou, X.-Q., & O'brien, J. L. (2012). Experimental realization of Shor's quantum factoring algorithm using qubit recycling. *Nature Photonics*, *6*(11), 773.

Microsoft. (2019). Quantum computing. Retrieved 2019-09-11, from https://www.microsoft.com/en-us/quantum/

Mitarai, K., Negoro, M., Kitagawa, M., & Fujii, K. (2018). Quantum circuit learning. *Physical Review A*, *98*(3), 032309.

Nielsen, M. A. & Chuang, I. L. (2011). *Quantum computation and quantum information: 10th anniversary edition* (10th). New York, NY, USA: Cambridge University Press.

Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., & Latorre, J. I. (2019). Data re-uploading for a universal quantum classifier. arXiv: 1907.02085 [quant-ph]

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., ... O'brien, J. L. (2014). A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, *5*, 4213.

Pradhan, S. S., Ward, W. H., Hacioglu, K., Martin, J. H., & Jurafsky, D. (2004). Shallow semantic parsing using support vector machines. In *Proceedings of the human language technology conference of the north american chapter of the association for computational linguistics: Hlt-naacl 2004* (pp. 233–240).

Preskill, J. (2012). Quantum computing and the entanglement frontier. arXiv: 1203.5813 [quant-ph]

Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum*, *2*, 79. doi:10.22331/q-2018-08-06-79

Raymer, M. G. & Monroes, C. (2019). The US national quantum initiative. *Quantum Science and Technology*, *4*(2), 020504. doi:10.1088/2058-9565/ab0441

Rebentrost, P., Mohseni, M., & Lloyd, S. (2014). Quantum support vector machine for big data classification. *Physical review letters*, *113*(13), 130503.

Romero, J. & Aspuru-Guzik, A. (2019). Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. arXiv: 1901.00848 [quant-ph]

Rønnow, T. F., Wang, Z., Job, J., Boixo, S., Isakov, S. V., Wecker, D., ... Troyer, M. (2014). Defining and detecting quantum speedup. *Science*, *345*(6195), 420–424.

Schuld, M., Bocharov, A., Svore, K., & Wiebe, N. (2018). Circuit-centric quantum classifiers. arXiv: 1804.00633 [quant-ph]

Schuld, M. & Killoran, N. (2019). Quantum machine learning in feature hilbert spaces. *Physical review letters*, *122*(4), 040504.

Schuld, M., Sinayskiy, I., & Petruccione, F. (2015). An introduction to quantum machine learning. *Contemporary Physics*, *56*(2), 172–185.

Shin, S. W., Smith, G., Smolin, J. A., & Vazirani, U. (2014). How "quantum" is the D-Wave machine? arXiv: 1401.7087 [quant-ph]

Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, *41*(2), 303–332.

SURFsara. (2019). The Lisa system. Retrieved 2019-09-20, from https://userinfo.surfsara.nl/systems/lisa